

Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Licenciatura en Ciencias Computacionales

3.2 SQL Fragmentos

Sexto semestre, Grupo dos

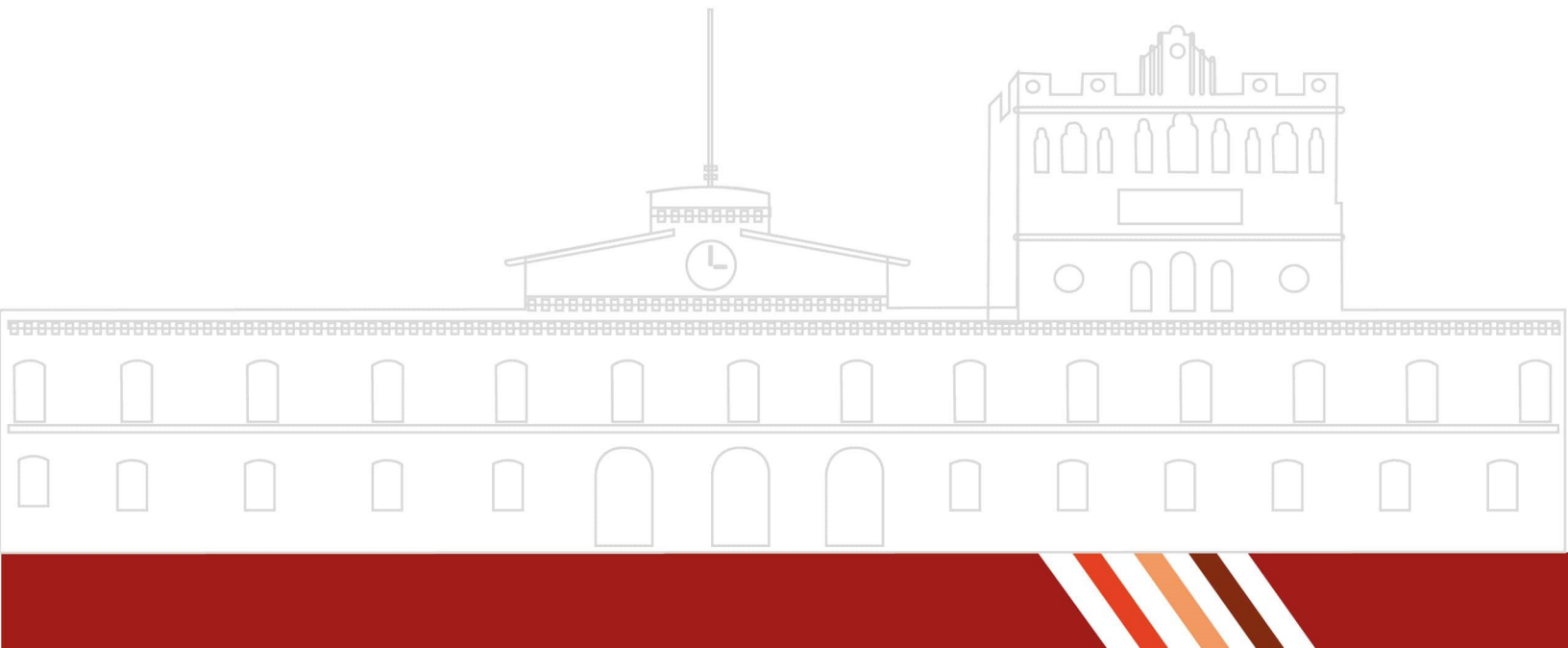
Catedrático: Dr. Eduardo Cornejo Velázquez

Base de datos distribuidas

Equipo:

Jennifer Resendiz Isidro

Juan Carlos Montes Gonzales



Contents

1	Introducción	3
2	Marco teórico	3
2.1	Fragmentación vertical	3
2.2	Procesos ETL	3
2.2.1	Fase de Extracción en los procesos ETL	3
2.2.2	Procesos ETL: fase de Transformación	3
2.2.3	Proceso de Carga: la culminación de los procesos ETL	4
2.3	SELECT + INTO FILE	4
2.3.1	Sintaxis	4
2.4	LOAD	4
2.4.1	Sintaxis	4
2.5	SELECT con tablas de dos bases de datos	5
2.5.1	Ejemplo	5
3	Herramientas empleadas	5
4	SQL Fragmentos	6
4.1	Esquema Conceptual Local de cada nodo.	6
4.2	Script de creación de nodos.	6
4.3	Script de extracción de datos.	9
4.4	Script de carga de datos	9
4.5	Script de consulta de datos a dos tablas en al menos dos de los nodos.	10
5	Conclusiones	10
6	Referencias Bibliográficas	11

Introducción

Práctica que consiste en usar la fragmentación y modificación de bases de datos distribuidas por medio de archivos.

Marco teórico

Fragmentación vertical

La fragmentación vertical se refiere a la división de una relación en subconjuntos de atributos (columna); cada subconjunto (fragmento) se guarda en un nodo diferente y cada fragmento tiene columnas únicas, con la excepción de la columna clave, la cual es común a todos los fragmentos. Esto es el equivalente de la sentencia `SELECT columna1, columna2 INTO NuevaTabla FROM Tabla`.

Procesos ETL

Los procesos ETL son una parte de la integración de datos, pero es un elemento importante cuya función completa el resultado de todo el desarrollo de la cohesión de aplicaciones y sistemas.

La palabra ETL corresponde a las siglas en inglés de:

- Extraer: extract.
- Transformar: transform
- Y Cargar: load.

2.2.1 Fase de Extracción en los procesos ETL

Para llevar a cabo de manera correcta el proceso de extracción, primera fase de los procesos ETL, hay que seguir los siguientes pasos:

1. Extraer los datos desde los sistemas de origen.
2. Analizar los datos extraídos obteniendo un chequeo.
3. Interpretar este chequeo para verificar que los datos extraídos cumplen la pauta o estructura que se esperaba. Si no fuese así, los datos deberían ser rechazados.
4. Convertir los datos a un formato preparado para iniciar el proceso de transformación

Además, uno de las prevenciones más importantes que se deben tener en cuenta durante el proceso de extracción sería el exigir siempre que esta tarea cause un impacto mínimo en el sistema de origen. Este requisito se basa en la práctica ya que, si los datos a extraer son muchos, el sistema de origen se podría ralentizar e incluso colapsar, provocando que no pudiera volver a ser utilizado con normalidad para su uso cotidiano.

2.2.2 Procesos ETL: fase de Transformación

La fase de transformación de los procesos de ETL aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Estas directrices pueden ser declarativas, pueden basarse en excepciones o restricciones pero, para potenciar su pragmatismo y eficacia, hay que asegurarse de que sean:

1. Declarativas.
2. Independientes.
3. Claras.

4. Inteligibles.
5. Con una finalidad útil para el negocio.

2.2.3 Proceso de Carga: la culminación de los procesos ETL

En esta fase, los datos procedentes de la fase anterior (fase de transformación) son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes.

Existen dos formas básicas de desarrollar el proceso de carga:

- Acumulación simple: esta manera de cargar los datos consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el data warehouse, almacenando un valor calculado que consistirá típicamente en un sumatorio o un promedio de la magnitud considerada. Es la forma más sencilla y común de llevar a cabo el proceso de carga.
- Rolling: este proceso sería el más recomendable en los casos en que se busque mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada (por ejemplo, totales diarios, totales semanales, totales mensuales, etc.).

SELECT + INTO FILE

Es un comando que exporta los resultados de una consulta SQL a un archivo externo, generalmente en formato texto o CSV, a diferencia de SELECT INTO que crea una nueva tabla en la base de datos. Esta instrucción es ideal para exportar datos para su análisis posterior, hacer copias de seguridad o moverlos a otros sistemas.

2.3.1 Sintaxis

```
1  SELECT columnas
2  FROM tabla
3  [WHERE condiciones]
4  INTO OUTFILE 'ruta/del/archivo.csv'
5  FIELDS TERMINATED BY ','
6  ENCLOSED BY '"'
7  LINES TERMINATED BY '\n';
```

LOAD

Lee filas de un archivo de texto en una tabla a gran velocidad. El archivo puede leerse desde el host del servidor o del cliente, dependiendo de si LOCAL se utiliza el modificador.

LOAD DATA Es el complemento de SELECT ... INTO OUT FILE.

2.4.1 Sintaxis

```
1  LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name.txt'
2  [REPLACE | IGNORE]
3  INTO TABLE tbl_name
4  [FIELDS
5   [TERMINATED BY '\t']
6   [[OPTIONALLY] ENCLOSED BY '']
```

```
7         [ESCAPED BY '\\'] ]
8     ]
9     [LINES
10         [STARTING BY '']
11         [TERMINATED BY '\\n']
12     ]
13     [IGNORE number LINES]
14     [(col_name,...)]
```

SELECT con tablas de dos bases de datos

La operación JOIN o combinación permite mostrar columnas de varias tablas como si se tratase de una sola tabla, combinando entre sí los registros relacionados usando para ello claves externas.

Las tablas relacionadas se especifican en la cláusula FROM, y además hay que hacer coincidir los valores que relacionan las columnas de las tablas.

2.5.1 Ejemplo

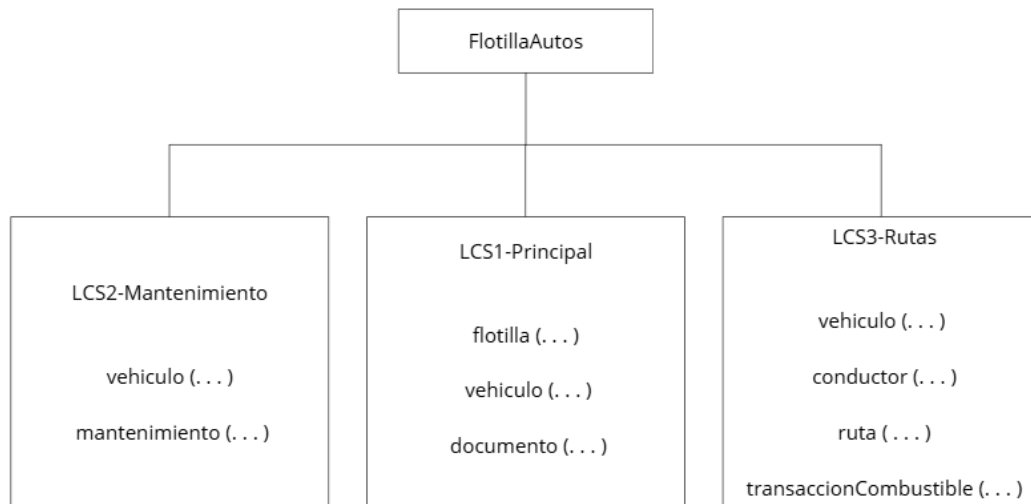
```
1 SELECT OrderID, C.CustomerID, CompanyName, OrderDate
2 FROM Customers C, Orders O
3 WHERE C.CustomerID = O.CustomerID;
```

Herramientas empleadas

1. **MySQL Server:** MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto. Se utiliza para almacenar, organizar y recuperar datos de manera eficiente, especialmente en aplicaciones web y sistemas de gestión de contenido (CMS). MySQL utiliza SQL (Structured Query Language) para interactuar con la base de datos, permitiendo a los usuarios realizar consultas, actualizaciones y otras operaciones sobre los datos [6].
2. **Overleaf:** Overleaf es un editor colaborativo de LaTeX en línea que permite a múltiples usuarios escribir, editar y publicar documentos científicos, técnicos y académicos de forma simultánea a través de un navegador web [9].

SQL Fragmentos

Esquema Conceptual Local de cada nodo.



Script de creación de nodos.

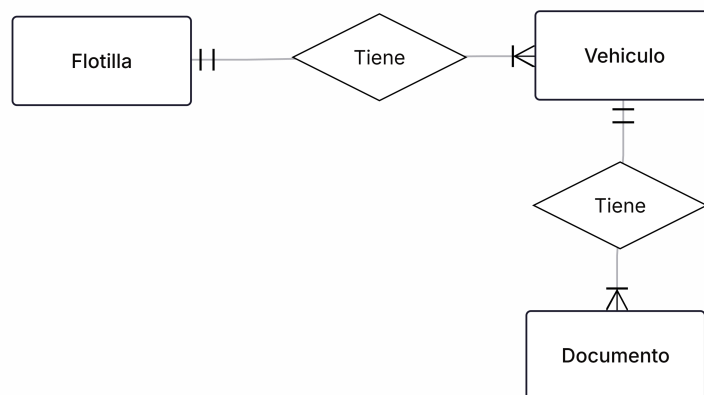


Figure 1: Principal

```
1 CREATE DATABASE LCS1_Principal;  
2 USE LCS1_Principal;  
3  
4 CREATE TABLE flotilla (  
5     id_flotilla INT NOT NULL,  
6     nombre VARCHAR(255) NOT NULL,
```

```

7      descripcion VARCHAR(500),
8      PRIMARY KEY(id_flotilla)
9  );
10
11 CREATE TABLE vehiculo (
12     id_auto INT NOT NULL,
13     marca VARCHAR(255) NOT NULL,
14     modelo VARCHAR(255) NOT NULL,
15     anio INT NOT NULL,
16     color VARCHAR(255) NOT NULL,
17     pasajeros INT NOT NULL,
18     placa VARCHAR(255) NOT NULL,
19     id_flotilla INT,
20     PRIMARY KEY(id_auto),
21     FOREIGN KEY (id_flotilla) REFERENCES flotilla(id_flotilla)
22 );
23
24 CREATE TABLE documento (
25     id_documento INT NOT NULL,
26     id_auto INT NOT NULL,
27     numero VARCHAR(255) NOT NULL,
28     nombre VARCHAR(255) NOT NULL,
29     inicio DATE NOT NULL,
30     final DATE NOT NULL,
31     monto DECIMAL(10,2) NOT NULL,
32     PRIMARY KEY(id_documento),
33     FOREIGN KEY (id_auto) REFERENCES vehiculo(id_auto)
34 );

```

Listing 1: Creación de nodo principal

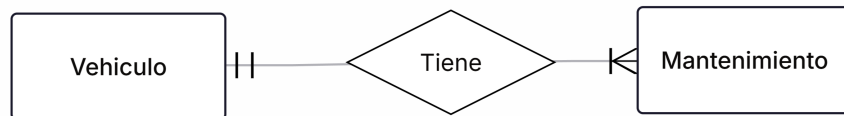


Figure 2: Mantenimiento

```

1 CREATE DATABASE LCS2_Mantenimiento;
2 USE LCS2_Mantenimiento;
3
4 CREATE TABLE vehiculo (
5     id_auto INT NOT NULL,
6     marca VARCHAR(255) NOT NULL,
7     modelo VARCHAR(255) NOT NULL,
8     anio INT NOT NULL,
9     color VARCHAR(255) NOT NULL,
10    placa VARCHAR(255) NOT NULL,
11    PRIMARY KEY(id_auto)
12 );
13
14 CREATE TABLE mantenimiento (
15     id_mantenimiento INT NOT NULL,
16     id_auto INT NOT NULL,
17     fechainicio DATE NOT NULL,

```

```

18  fechafinal DATE NOT NULL,
19  diagnostico VARCHAR(255) NOT NULL,
20  descripcion VARCHAR(500) NOT NULL,
21  PRIMARY KEY(id_mantenimiento),
22  FOREIGN KEY (id_auto) REFERENCES vehiculo(id_auto)
23 );

```

Listing 2: Creación de nodo mantenimiento

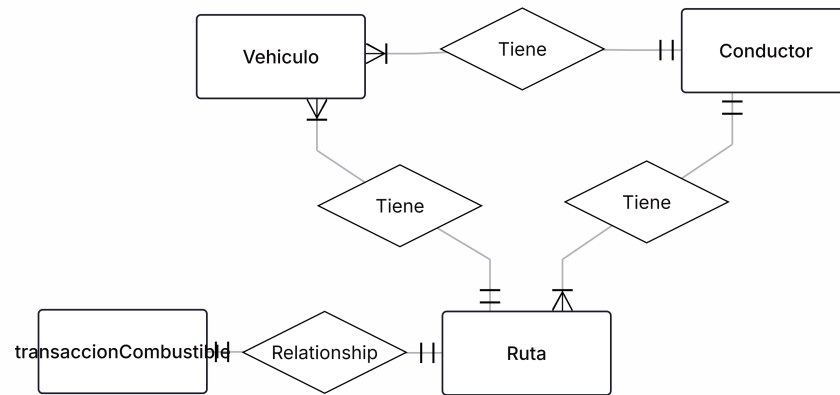


Figure 3: Rutas

```

1  CREATE DATABASE LCS3_Rutas;
2  USE LCS3_Rutas;
3
4  CREATE TABLE vehiculo (
5      id_auto INT NOT NULL,
6      marca VARCHAR(255) NOT NULL,
7      modelo VARCHAR(255) NOT NULL,
8      placa VARCHAR(255) NOT NULL,
9      PRIMARY KEY(id_auto)
10 );
11
12 CREATE TABLE conductor (
13     id_conductor INT NOT NULL,
14     nombre VARCHAR(255) NOT NULL,
15     apellidopaterno VARCHAR(255) NOT NULL,
16     apellidomaterno VARCHAR(255) NOT NULL,
17     telefono VARCHAR(255) NOT NULL,
18     licencia VARCHAR(255) NOT NULL,
19     id_auto INT NOT NULL, -- Columna agregada para la FK
20     PRIMARY KEY(id_conductor),
21     FOREIGN KEY (id_auto) REFERENCES vehiculo(id_auto)
22 );
23
24 CREATE TABLE ruta (
25     id_ruta INT NOT NULL,
26     id_auto INT NOT NULL,
27     id_conductor INT NOT NULL,
28     fecha DATE NOT NULL,
29     horainicio TIME NOT NULL,

```



```

30     origen VARCHAR(255) NOT NULL,
31     destino VARCHAR(255) NOT NULL,
32     horallegada TIME NOT NULL,
33     cobro DECIMAL(10,2) NOT NULL,
34     PRIMARY KEY(id_ruta),
35     FOREIGN KEY (id_auto) REFERENCES vehiculo(id_auto),
36     FOREIGN KEY (id_conductor) REFERENCES conductor(id_conductor)
37 );
38
39 CREATE TABLE transaccionCombustible (
40     id_transaccion INT NOT NULL,
41     id_ruta INT NOT NULL,
42     litros DECIMAL(10,2) NOT NULL,
43     costo DECIMAL(10,2) NOT NULL,
44     fecha DATE NOT NULL,
45     PRIMARY KEY(id_transaccion),
46     FOREIGN KEY (id_ruta) REFERENCES ruta(id_ruta)
47 );

```

Listing 3: Creación de nodo rutas

Script de extracción de datos.

```

1 SELECT 'id_auto', 'marca', 'modelo', 'anio', 'color', 'pasajeros', 'placa', '
   id_flotilla'
2 UNION ALL
3 SELECT id_auto, marca, modelo, anio, color, pasajeros, placa, id_flotilla
4 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculos.csv'
5 FIELDS TERMINATED BY ';'
6 ENCLOSED BY ' '
7 LINES TERMINATED BY '\n'
8 FROM LCS1_Principal.vehiculo;

```

Listing 4: Scripts de extracción de datos

```

mysql> SELECT 'id_auto', 'marca', 'modelo', 'anio', 'color', 'pasajeros', 'placa', 'id_flotilla'
-> UNION ALL
-> SELECT id_auto, marca, modelo, anio, color, pasajeros, placa, id_flotilla
-> INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/docs/vehiculos.csv'
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY ' '
-> LINES TERMINATED BY '\n'
-> FROM LCS1_Principal.vehiculo;

```

Figure 4: Captura

Script de carga de datos

```

1 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculos.csv'
2 IGNORE INTO TABLE LCS1_Principal.vehiculo
3 FIELDS TERMINATED BY ';'
4 ENCLOSED BY ' '
5 LINES TERMINATED BY '\n'
6 IGNORE 1 LINES
7 (id_auto, marca, modelo, anio, color, pasajeros, placa, id_flotilla);

```

Listing 5: Carga de datos

```
mysql> LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculos.csv'
-> IGNORE INTO TABLE LCS1_Principal.vehiculo
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (id_auto, marca, modelo, anio, color, pasajeros, placa, id_flotilla);
Query OK, 0 rows affected, 5 warnings (0.00 sec)
```

Figure 5: Ejecución

id_auto	marca	modelo	anio	color	pasajeros	placa	id_flotilla
1	Toyota	Corolla	2023	Blanco	5	ABC-123	1
2	Nissan	Urvan	2022	Azul	12	DEF-456	4
3	Mercedes	Benz	2024	Negro	4	GHI-789	3
4	Chevrolet	Aveo	2023	Rojo	5	JKL-012	2
5	Ford	Transit	2022	Blanco	15	MNO-345	4

Figure 6: Captura archivo de datos

Script de consulta de datos a dos tablas en al menos dos de los nodos.

Consulta: Ver el mantenimiento más reciente de los vehículos que realizaron rutas

```
1 SELECT
2     v1.id_auto,
3     v1.marca,
4     v1.modelo,
5     v1.placa,
6     v1.anio,
7     (SELECT COUNT(*) FROM LCS2_Mantenimiento.mantenimiento m
8      WHERE m.id_auto = v1.id_auto) as total_mantenimientos,
9     (SELECT MAX(m.fechafinal) FROM LCS2_Mantenimiento.mantenimiento m
10     WHERE m.id_auto = v1.id_auto) as ultimo_mantenimiento
11 FROM LCS1_Principal.vehiculo v1
12 ORDER BY v1.id_auto;
```

Listing 6: Consultas

id_auto	marca	modelo	placa	anio	total_mantenimientos	ultimo_mantenimiento
1	Toyota	Corolla	ABC-123	2023	1	2024-01-12
2	Nissan	Urvan	DEF-456	2022	1	2024-01-17
3	Mercedes	Benz	GHI-789	2024	1	2024-01-20
4	Chevrolet	Aveo	JKL-012	2023	1	2024-01-25
5	Ford	Transit	MNO-345	2022	1	2024-01-28

Figure 7: Consulta

Conclusiones

Esta práctica permitió poner a prueba nuestras habilidades para fragmentación de bases de datos distribuidas.

Referencias Bibliográficas

References

- [1] Datacamp.com. (2025, June 4). *Fragmentación de bases de datos: Ejemplos, estrategias, herramientas y más*. <https://www.datacamp.com/es/blog/database-sharding>
- [2] Edenred, E. (2022, diciembre 2). *Flotilla de autos: cómo administrarla*. Edenred.mx. <https://www.edenred.mx/blog/flotilla-de-autos-como-administrarla>
- [3] Instituto Consorcio Clavijero. (n.d.). *Tema 3.2 Fragmentación de la base de datos - Bases de datos distribuidas*. Cursos.clavijero.edu.mx. https://cursos.clavijero.edu.mx/cursos/080_bdd/modulo3/contenidos/tema3.2.html
- [4] Hightouch. (2023, December 29). *SQL SELECT INTO - Syntax, Use Cases, and Examples*. Hightouch.com. <https://hightouch.com/sql-dictionary/sql-select-into>
- [5] Nust.na. (2025). *MySQL :: MySQL 5.0 Reference Manual :: 13.2.5 Sintaxis de LOAD DATA INFILE*. <http://download.nust.na/pub6/mysql/doc/refman/5.0/es/load-data.html>
- [6] Erickson, J. (2024, August 29). *MySQL: Understanding what it is and how it's used*. Oracle. <https://www.oracle.com/mx/mysql/what-is-mysql/#:~:text=MySQL%20es%20un%20sistema%20de,la%20larga%20popularidad%20de%20MySQL>
- [7] PowerData, R. (n.d.). *¿Qué son los procesos ETL?* Blog.powerdata.es. <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/qu-son-los-procesos-etl>
- [8] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2006). *Fundamentos de bases de datos* (5a ed., F. Sáenz Pérez, A. García Cordero & J. Correas Fernández, Trads.). McGraw-Hill/Interamericana de España.
- [9] Universidad Americana De Europa. (2025, August 4). *Overleaf: Herramienta definitiva para escribir en LaTeX online*. <https://unade.edu.mx/overleaf-herramienta-definitiva-para-escribir-en-latex-online/>